J-M Moreno

Mardi 27 mars 2018

Attention

Sauf indication contraire, les questions sont indépendantes. Vous devez expliciter et argumenter vos réponses. Par ailleurs il n'y a pas forcément de « bonne » réponse, ou même de réponse, l'argumentation que vous adosserez à vos commentaires n'en sera que plus importante.

1 Vous trouverez dans ce qui suit, le résultat de l'exécution de quelques commandes que nous allons détailler. Tout d'abord `ls -lRh` donne la liste de façon récursive d'un catalogue, en affichant le résultat d'une manière facilement lisible — « human ». Ensuite `df -kh` affiche la taille totale et celle des espaces libre et occupé d'une partition.

```
<brumaire-1304-[10:25]>% ls -lRh /essai
total 2756992
-rwxrwxr-x 1 jmm wheel 2.6G Mar 23 10:25 fichier
<brumaire-1305-[10:26]>% df -kh /essai
Filesystem      Size      Used      Avail Capacity  Mounted on
/dev/da1p6      9.7G      4.4G      4.5G      49%      /essai
<brumaire-1306-[10:26]>% rm /essai/fichier
<brumaire-1307-[10:27]>% ls -lRh /essai
total 0
<brumaire-1308-[10:27]>% df -kh /essai
Filesystem      Size      Used      Avail Capacity  Mounted on
/dev/da1p6      9.7G      7.5G      1.4G      84%      /essai
<brumaire-1309-[10:27]>%
```

Le but de `ls -lRh /essai` est donc de montrer que le partition montée sur `/essai` ne contient qu'un seul fichier nommé `fichier`. Après l'exécution de `rm /essai/fichier`, la partition `/essai` est totalement vide. Que remarquez vous entre les deux exécutions de `df -kh /essai`. Si vous remarquez quelque chose d'étrange qu'elle peut en être la raison ?

2 Vous trouverez ci-après un programme C relativement simple et trivial. Il s'agit d'une simple boucle infinie, n'y cherchez aucun sens particulier.

```
#include <stdio.h>
#include <fcntl.h>
unsigned long long tableau[4210000000];
main()
{
    while (1) {
        tableau[0]++;
        tableau[0]--;
    }
}
```

Il vous est simplement demandé de prévoir son impact sur les ressources du système. Pour vous faciliter la tâche sachez que le tableau a une taille d'environ 320 Go. Afin de vous aider vous trouverez ci-dessous la résultat de l'exécution de la commande *ps*, le processus *gros* correspondant évidemment au programme ci-dessus.

```
<brumaire-1093-[16:46]>% ps -O vsz,rss -p 74005
  PID      VSZ  RSS TT  STAT      TIME COMMAND
74005 32896888 2020 0 R   169:20.90 ./gros
<brumaire-1094-[16:46]>%
```

Les options *-O vsz,rss* et *-p 74005* servent ici à réduire l'affichage aux informations qui nous intéressent. Les champs *TT* et *STAT* ne jouent aucun rôle ici, vous pouvez les ignorer. Le champ *VSZ* donne la taille virtuelle et *RSS* la taille résidente¹.

3 Il vous est demandé ici de vous interroger de façon similaire sur le programme *C* ci-dessous. Encore une fois ne cherchez aucun sens caché à ce programme, il n'a qu'un simple but de démonstration dans la cadre de l'exercice.

Comme précédemment donnez l'impact que va pouvoir avoir son exécution sur le système. Pour rappel la primitive *pause()* provoque une attente de signal.

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
unsigned long long tableau[421000000];
main()
{
    while (1) {
        pause();
    }
}
```

Regardez bien attentivement le code du programme. Il serait particulièrement judicieux de vous interroger sur l'utilité de la boucle infinie.

4 Nous allons prendre le problème à l'envers. Vous trouverez ci-dessous l'affichage du résultat d'une commande *ps*. Comme précédemment vous pouvez ignorer les champs *TT* et *STAT*. Le champ *MWCHAN* donne le nom de l'évènement que le processus attend, c'est-à-dire le plus souvent l'appel système en cours de réalisation.

```
<brumaire-1152-[11:21]>% ps -O vsz,rss,mwchan -p 76829
  PID  VSZ  RSS MWCHAN TT  STAT      TIME COMMAND
76829 6264 2020 nanslp 1  S   0:00.00 ./programme
<brumaire-1153-[11:22]>%
```

Dites, si d'après vous, le processus *programme* correspond à l'exécution d'un des deux programmes précédents. Dans tous les cas expliquez pourquoi.

La valeur *nanslp* du champ *MWCHAN* indique un appel à la primitive *nanosleep()* elle-même appelée par la fonction *sleep()* dont vous vous imaginez le rôle.

5 Vous trouverez ici le résultat de l'exécution des commandes *top* et *sysctl*. L'option *-b 1* de *top* permet de limiter l'affichage à une seule ligne. La commande *sysctl* permet, ici, d'extraire la valeur de paramètres issus de la configuration du noyau. Dans le cas présent on a requis le modèle des processeurs, leur architecture et leur nombre — on obtient en fait le nombre de cœurs ce qui revient au même ici.

Le dernier paramètre, *hw.physmem*, correspond à la mémoire physique intégrée à la machine — le nombre de « barrettes » si vous préférez, cette valeur est donnée en octets.

¹ Il sont équivalents aux champs *SIZE* et *RES*, respectivement de la commande *top*.

```

<brumaire-1201-[15:10]>% top -b 1
last pid: 77410; load averages:  9.00,  9.00,  8.93  up 2+04:14:01   15:10:47
39 processes: 10 running, 28 sleeping, 1 stopped
Mem: 3584K Active, 217M Inact, 2140M Wired, 806M Buf, 150G Free
Swap: 128G Total, 128G Free
  PID USERNAME   THR PRI NICE   SIZE   RES STATE  C  TIME   WCPU COMMAND
74005 jmm           1 103   0 32126M 2020K CPU7   7 25.2H 100.00% gros
<brumaire-1202-[15:10]>% sysctl hw.model hw.machine hw.ncpu hw.physmem
hw.model: AMD Opteron(tm) Processor 6164 HE
hw.machine: amd64
hw.ncpu: 24
hw.physmem: 167468470272

```

À partir de ces différents éléments examinez et commentez les différentes caractéristiques de la machine. Pour information sachez que 167468470272 correspond à 156 Go. Le modèle du processeur vous est donné à titre d'information il n'a pas grande importance dans le cas présent.

6 Dans ce qui suit vous trouverez l'affichage correspondant à une exécution de la commande *ifconfig*. Pour rappel, *ifconfig*, permet soit de configurer une interface réseau, soit d'afficher ses caractéristiques. C'est le cas ici.

```

root@huile-frelatee:~ # ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  ether 68:b5:99:b6:b0:16
  inet 194.254.199.43 netmask 0xffffffff broadcast 194.254.199.255
  inet 10.70.70.43 netmask 0xffff0000 broadcast 10.70.255.255
  inet 192.168.79.43 netmask 0xfffff00 broadcast 192.168.79.255
  inet 192.168.78.43 netmask 0xfffff00 broadcast 192.168.78.255
  inet6 fe80::6ab5:99ff:feb6:b016%lagg0 prefixlen 64 scopeid 0x6
  inet6 2001:660:3301:8070::43 prefixlen 64
  media: Ethernet autoselect
  status: active
  groups: lagg
  laggport: igb0 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
  laggport: igb1 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
  laggport: igb2 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
  laggport: igb3 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
root@huile-frelatee:~ #

```

Certains éléments de l'affichage qui n'avaient aucune importance ici ont été supprimés afin de l'alléger. Veuillez commenter le résultat. Que pouvez vous dire des différentes adresses ? Que sont les deux adresses IPv6, marquées *inet6*. Pouvez vous dire à quoi correspondent *igb0*, *igb1*, *igb2*, *igb3* ? Qu'est d'après vous l'interface *lagg0* ?

7 On étudie ici un second affichage correspondant à une exécution de la commande *ifconfig*. Comme précédemment quelques éléments sans importance on été supprimés.

```

root@huile-frelatee:~ # ifconfig vlan929
vlan929: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  ether 68:b5:99:b6:b0:16
  inet 192.168.70.43 netmask 0xfffff00 broadcast 192.168.70.255
  media: Ethernet autoselect
  status: active
  vlan: 929 vlanpcp: 0 parent interface: lagg0
  groups: vlan
root@huile-frelatee:~ #

```

D'après vous à quoi correspond cette interface *vlan929* ? Celle-ci a t'elle un lien avec celle décrite précédemment ? Dans tous les cas donnez les arguments qui conditionnent votre réponse.

8 On décide d'acquérir un serveur pour une application de type *AMP*, c'est-à-dire *Apache*, *Mysql* et *PHP*. Autrement dit une application *WEB* intégrant une base de données. Donnez les spécifications qui seraient, selon vous, nécessaires à ce serveur. C'est-à-dire les caractéristiques en termes de processeur, de mémoire et de disque. Argumentez votre réponse.